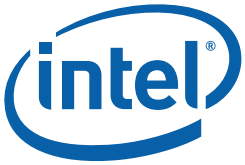


Vertex Processing Selection Capability

White Paper for Intel® G965, G35, and GM965 Express Chipsets

December 2007

Revision 001

Document Number: 318888-001

Contents

Figures

Revision History

Revision Number	Description	Revision Date
-001	Initial release.	December 2007

§

1 Introduction

The purpose of this document is to explain what the vertex processing selection capability is and what the benefits are. The document is targeted for OEM's, system integrators, and end-users that will be building or using motherboards with Intel® G965, GM965, and G35 Express Chipsets.

1.1 Terminology

Term	Description
TnL	Transform and light
PSGP	Processor Specific Graphics Pipeline: Code path that enables Vertex processing and TnL (transform and light) calculation to run on a processor.
SWVP	Software Vertex Processing: Vertex Processing that takes place on the CPU utilizing the PSGP
HWVP	Hardware Vertex Processing: Vertex Processing that takes place on a graphics chip

§

2 Section 2

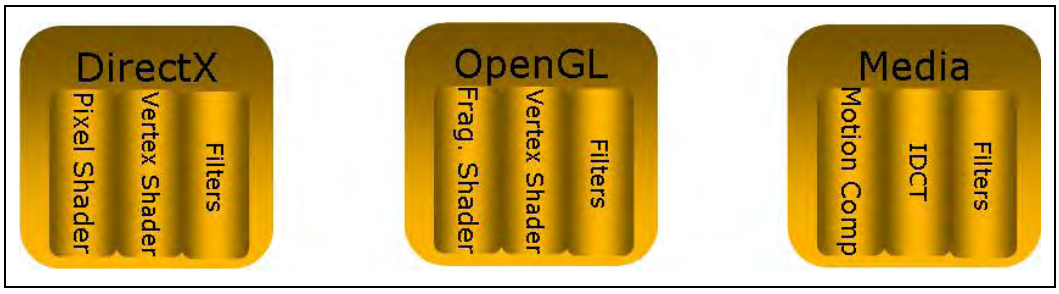
2.1 Introduction

Intel recently introduced the 15.6 and 14.31 Windows Vista* and Windows* XP graphics drivers that enables Shader Model 3.0 including support for hardware vertex shader and HW TnL on the Intel® G965, GM965, and G35 Express Chipsets. This capability has shown enhancements in game compatibility as well as game play. Along with this capability, Intel also introduced a vertex processing selection capability which allows the drivers to switch between using the integrated graphics engine and the Intel processors for vertex processing depending on the application. The end result is that Intel is able to deliver the highest possible frame rates by leveraging Intel's world class processors.

2.2 Integrated Graphics Architecture

The Intel GMA X3x00 architecture is based upon a unified shader architecture that is comprised of a collection of highly programmable and scalable 32-bit floating point engines. This programmable architecture is one based upon a unified shader model, which means both vertex shaders and pixels shaders run on the same programmable engines. These same programmable engines are also able to process media codecs, as well as various filters for text and image processing. The Intel graphics driver architecture delivers scalability and flexibility by supporting features within kernels that are compiled for the Intel graphics architecture. At the high level there are kernels that support 3D and media. Within 3D there are kernels for DirectX* and OpenGL*. DirectX* then has kernels that support pixel and vertex shading, and filtering. OpenGL has fragment and vertex shading, as well as filtering kernels. The media kernels are codec dependent, but capabilities like IDCT (Inverse Discrete Cosine Transform), motion compensation, and filtering techniques like advanced de-interlacing would all be supported by a kernel. Figure 1 is a view of the driver architecture with kernels.

Figure 1: 3D and Media Kernels



2.3 Why Vertex Processing Selection Capability?

In the process of designing and enabling a driver that supports Vertex Processing and TnL, Intel engineers discovered that some applications, mainly 3D games, performed better with vertex processing done on the processor rather than on the graphics engine. As mentioned above, the architecture uses the same programmable engines to process all shaders. By off-loading some of the vertex shader processing to the processor, the graphics engine is able to process more pixel shader data and do additional work like anisotropic filtering. Therefore, the graphics engine is able to do more while staying within market required power, thermal, and cost envelopes.

Separate vertex shader and pixel shader units within the graphics engine was a potential design option for Intel graphics engines. However, upon analysis of various 3D applications, it was discovered that the vertex traffic is at the beginning of each scene, and only accounted for one-third of the compute requirements for a scene. On the other hand, pixel traffic is about two-thirds of the scene and required most of the compute resources. So for much of the time the vertex shaders would be sitting idle. Of course this is application specific and one we will explore further. Figure 2 shows a small sample of applications that perform better with Vertex Processing being done on our integrated graphics parts. As one can see Age of Empires 3* receives a modest gain of 15%, while Far Cry* and Hal-Life 2 Episode 1* see extremely large gains of 95% and 230% respectively. Figure 3 shows a small sample of applications and the benefits of software vertex processing. When using software vertex processing, the performance benefit gained over hardware vertex processing ranges from 30-65% across a small sample of games. A 30% performance benefit was measured for FEAR*, 60% for Guild Wars* and 65% for Grand Theft Auto*. See Figure 3 below.

Figure 2: Performance with HWVP

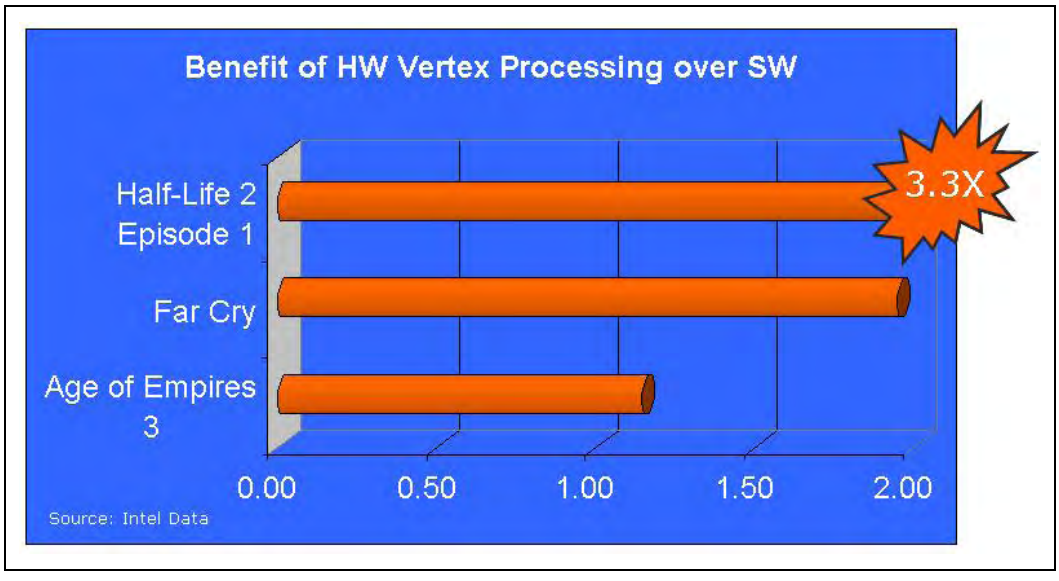
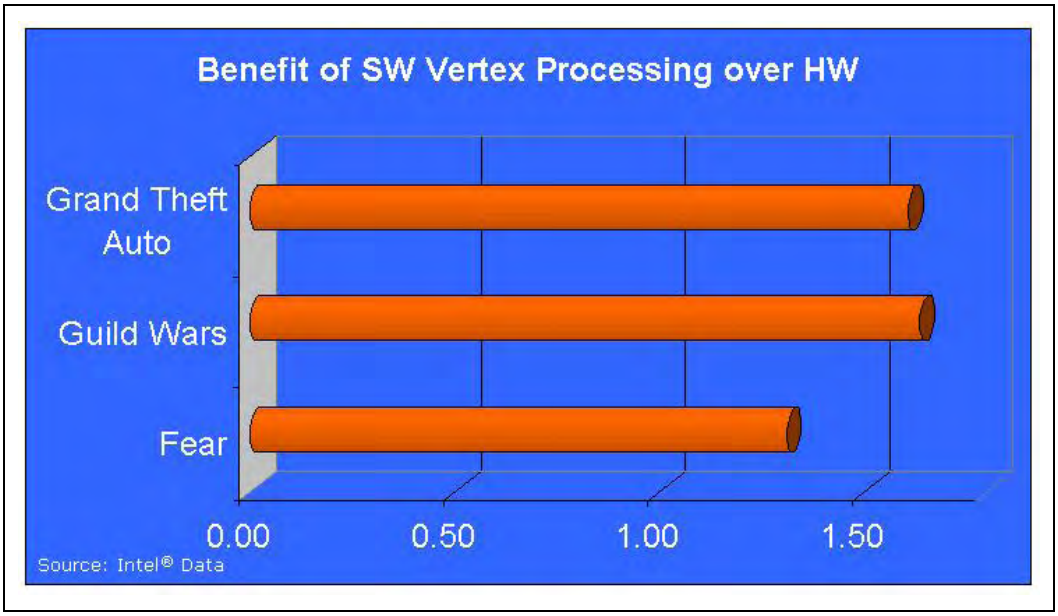


Figure 3: Benefits of Vertex Processing Selection Capability (SWVP)



2.4 Driver Architecture for Vertex Processing Selection Capability

Within any given 3D API there are certain capabilities that are exposed to the application. The Intel graphics driver always exposes Vertex Shader 3.0, and below, as well as HW TnL capabilities bits by default. Many applications perform best when the vertex processing is done on Intel graphics engines. As stated above, there are several exceptions where the application may perform better when vertex processing is done on a processor. If the application performs better with processor-based vertex processing, and has met Intel's selection criteria, it is placed in a list of applications contained within the driver INF file. The driver detects the application executables, turns off the integrated graphics vertex processing path and turns on the PSGP path that runs on an Intel processor. Figure 4 is a block diagram showing the graphics engine path (HWVP). Figure 5 is a block diagram showing the processor path (SWVP).

Applications are validated across different configurations and results are reviewed to determine which applications qualify for inclusion into the PSGP list. As new applications are launched, they are reviewed for inclusion on the list as shown below in Figure 6 (a list of 3D applications that are supported by the PSGP (SWVP)).

Figure 4: Block Diagram Showing the Graphics Engine Path - HWVP

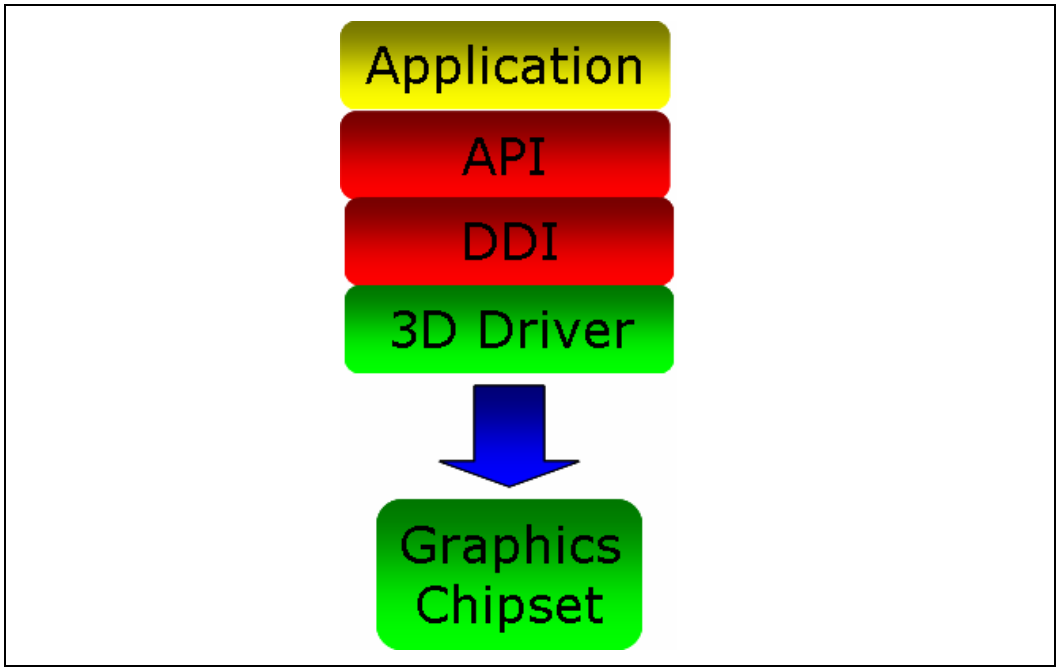


Figure 5: Block Diagram Showing the Processor Path - SWVP

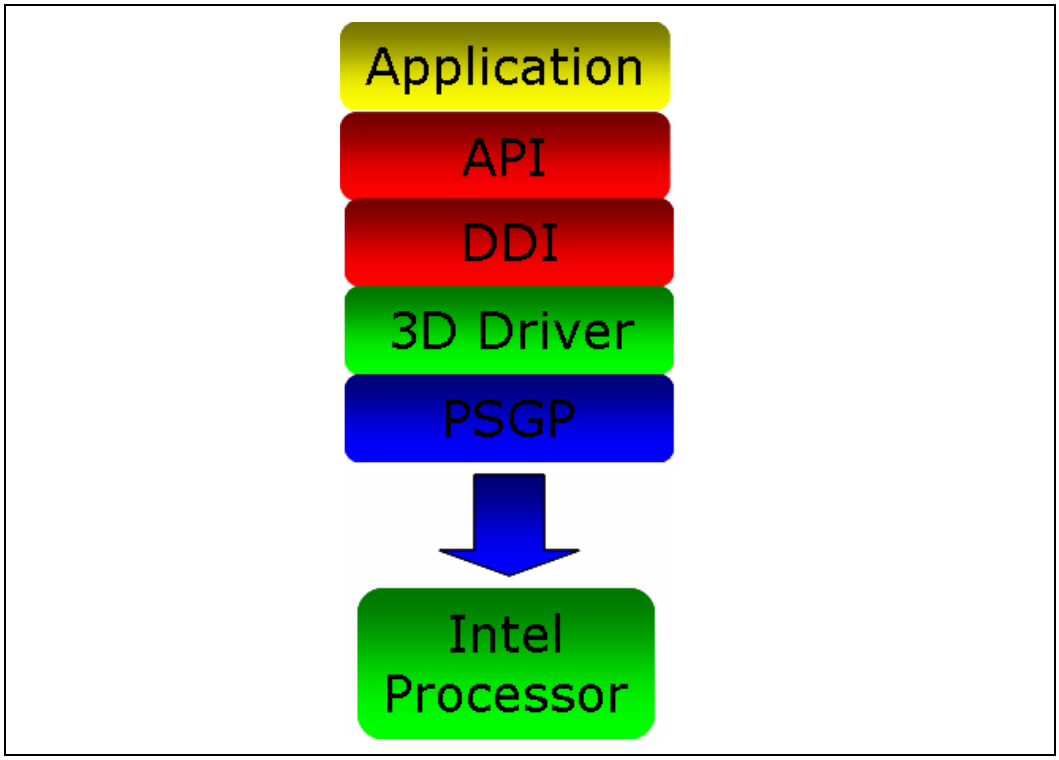


Figure 6: 3D Applications Supported by SWVP

XP	Vista
3DMark06	3DMark06
3DMark05	3DMark05
3DMark03	3DMark03
PCMark05	PCMark05
Serious Sam 2	Serious Sam 2
GTA: San Andreas	GTA: San Andreas
FEAR	FEAR
FFXiWinBench	FFXiWinBench
WoW	WoW
Company of Heroes	Company of Heroes
AOE3	AOE3
Madden 07	Madden 07
	STALKER
	UT2004
	Guild Wars

NOTE: Other names and brands may be claimed as the property of others.

2.5 Summary

Intel recently introduced graphic driver versions 15.6 and 14.31 for Windows Vista and Windows XP that enable Shader Model 3.0 (including support for hardware vertex shader and HW TnL on the Intel® G965, GM965, and G35 Express Chipsets.) Intel also introduced the capability to switch between HWVP and SWVP based upon the 3D application running on Intel integrated graphics engines. Results showed that some 3D applications perform best with HWVP, which is the default configuration. In other instances, results showed that some 3D applications perform best with SWVP. Intel's goal is to provide the best user experience possible for 3D applications while leveraging the performance of the processor where it makes sense.